

TITLE: **System and Method for Enabling a Client Application to Operate
Offline from a Server**

INVENTOR: **Theron Tock**

APPENDIX A

```
// action.idl : IDL source for action.dll
//
```

```
// This file will be processed by the MIDL tool to
// produce the type library (action.tlb) and marshalling code.
```

```
import "oaidl.idl";
import "ocidl.idl";
```

```
typedef enum tagHSAActionResult
{
    HSACTION_SUCCEEDED,
    HSACTION_FAILED,
    HSACTION_RETRY,
    HSACTION_STATUS_UNKNOWN
} HSAActionResult;
```

```
{
    object,
    uuid(E756E9A0-CFCA-11D1-ADAF-00609724000B),
    dual,
    helpstring("IHSActionScheduler Interface"),
    pointer_default(unique)
}
```

```
interface IHSActionScheduler : IDispatch
```

```
{
    [id(1), helpstring("method ScheduleAction")] HRESULT ScheduleAction(
        [in] long objectId,
        [in] const BSTR callbackObjectName,
        [in] long priority,
        [in] BOOL forCurrentUserOnly,
        [in] BOOL exactlyOnceSemantics);
    [id(2), helpstring("method CancelAction")] HRESULT CancelAction(
        [in] long objectId,
        [in] const BSTR callbackObjectName);
    [id(3), helpstring("method WaitForAction")] HRESULT WaitForAction(
        [in] long objectId,
        [in] const BSTR callbackObjectName);
    [id(4), helpstring("method RegisterSynchronizers")] HRESULT RegisterSynchronizers(
        [in] BSTR synchronizers);
    [id(5), helpstring("method RunAllSynchronizers")] HRESULT RunAllSynchronizers(
        [in] VARIANT_BOOL waitForCompletion);
    [id(6), helpstring("method CancelActionsForObject")] HRESULT CancelActionsForObject(
        [in] BSTR callbackName);
    [propget, id(7), helpstring("property DatabaseName")] HRESULT DatabaseName([out, ret
val] BSTR *pVal);
    [propput, id(7), helpstring("property DatabaseName")] HRESULT DatabaseName([in] BSTR
newVal);
    [id(8), helpstring("method ResetSynchronizers")] HRESULT ResetSynchronizers([in] BST
R synchronizers);
    [id(9), helpstring("method ResetAllSynchronizers")] HRESULT ResetAllSynchronizers();
};
```

```
// You implement an object that supports this interface if you want
// to be scheduled
```

```
{
    object,
    uuid(E756E9A1-CFCA-11D1-ADAF-00609724000B),
    dual,
```

```

    helpstring("IHSAction Interface"),
    pointer_default(unique)
}
interface IHSAction : IDispatch
{
    [id(1), helpstring("method DoAction")] HRESULT DoAction(
        [in] long objectId,
        [in] long sequenceId,
        [out, retval] HSActionResult *succeeded);
};

// You implement an object that supports this interface if you want to
// synchronize
{
    object,
    uuid(D26770E3-F5E7-11d1-ADD2-00609724000B),
    dual,
    helpstring("IHSSynchronize Interface"),
    pointer_default(unique).
}
interface IHSSynchronize : IDispatch
{
    [id(1), helpstring("method DoSync")] HRESULT DoSync([out, retval] long *minutesUntil
NextSync);
};

[
    uuid(E756E993-CFCA-11D1-ADAF-00609724000B),
    version(1.0),
    helpstring("action 1.0 Type Library")
]
library HSACTION
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid(2AE0043B-C911-11D1-ADA0-00609724000B),
        helpstring("HSActionScheduler Class")
    ]
    coclass HSActionScheduler
    {
        [default] interface IHSActionScheduler;
    };
};

```